TECHNICAL RESEARCH REPORT

An Adaptive Sampling Algorithm for Solving Markov Decision Processes

*by Hyeong Soo Chang, Michael C. Fu,
and Steven I. Marcus*

TR 2002-19

ISR
INSTITUTE FOR SYSTEMS RESEARCH

ISR develops, applies and teaches advanced methodologies of design and analysis to solve complex, hierarchical, heterogeneous and dynamic problems of engineering technology and systems for industry and government.

ISR is a permanent institute of the University of Maryland, within the Glenn L. Martin Institute of Technology/A. James Clark School of Engineering. It is a National Science Foundation Engineering Research Center.

Web site http://www.isr.umd.edu

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **2002**	2. REPORT TYPE	3. DATES COVERED **-**

4. TITLE AND SUBTITLE
An Adaptive Sampling Algorithm for Solving Markov Decision Processes

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research,875 North Randolph Street,Arlington,VA,22203-1768

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**17**	

An Adaptive Sampling Algorithm for Solving Markov Decision Processes

Hyeong Soo Chang, Michael C. Fu, and Steven I. Marcus
Institute for Systems Research
University of Maryland, College Park, MD 20742
E-mail: {hyeong,mfu,marcus}@isr.umd.edu

May 31, 2002

Abstract

Based on recent results for multi-armed bandit problems, we propose an adaptive sampling algorithm that approximates the optimal value of a finite horizon Markov decision process (MDP) with infinite state space but finite action space and bounded rewards. The algorithm adaptively chooses which action to sample as the sampling process proceeds, and it is proven that the estimate produced by the algorithm is asymptotically unbiased and the worst possible bias is bounded by a quantity that converges to zero at rate $O\left(\frac{H \ln N}{N}\right)$, where H is the horizon length and N is the total number of samples that are used per state sampled in each stage. The worst-case running-time complexity of the algorithm is $O((|A|N)^H)$, independent of the state space size, where $|A|$ is the size of the action space. The algorithm can be used to create an approximate receding horizon control to solve infinite horizon MDPs.

Keywords: (adaptive) sampling, Markov decision process, multi-armed bandit problem, receding horizon control

1 Introduction

In this paper, we propose an "adaptive" sampling algorithm that approximates the optimal value to break the well-known *curse of dimensionality* in solving finite horizon Markov decision processes (MDPs). The algorithm is aimed at solving MDPs with a large (possibly infinite) state space but with a finite action space and bounded rewards. The approximate value computed by the algorithm not only converges to the true optimal value but also does so in an "efficient" way. The algorithm adaptively chooses which action to sample as the sampling process proceeds and the estimate produced by the algorithm is asymptotically unbiased and the worst possible bias is bounded by a quantity that converges to zero at rate[1] of $O\left(\sum_{i=1}^{H} \frac{\ln N_i}{N_i}\right)$, where H is the length of the horizon and N_i is the total number of samples which are used per state sampled in stage i. The logarithmic bound in the numerator is achievable uniformly over time. Given that the action space size is $|A|$, the worst-case running time-complexity of the algorithm is $O\left((|A| \max_{i=1,\ldots,H} N_i)^H\right)$, which is independent of the state space size but is dependent on the size of the action space due to the requirement that each action be sampled at least once at each sampled state

The idea behind the adaptive sampling algorithm is based on the expected *regret* analysis of the multi-armed bandit problem developed by Lai and Robbins (1985). In particular, we exploit the recent finite-time analysis work by Auer, Cesa-Bianchi, and Fischer (2002) that elaborated Agrawal (1995). The goal of the multi-armed bandit problem is to play as often as possible the machine that yields the highest (expected) reward. The regret quantifies the exploration/exploitation dilemma in the search for the true "optimal" machine, which is unknown in advance. During the search process, we wish to explore the reward distribution of different machines while also frequently playing the machine that is empirically best thus far. The regret is the expected loss due to not always playing the true optimal machine. Lai and Robbins (1985) showed that for an optimal strategy the regret grows at least logarithmically in the number of machine plays, and recently Auer, Cesa-Bianchi, and Fischer (2002) showed that the logarithmic regret is also achievable uniformly over time with a simple and efficient sampling algorithm for arbitrary reward distributions with bounded support. We incorporate their results into a sampling-based process for finding an optimal action in a state for a single stage of an MDP by appropriately converting the definition of regret into the difference between the true optimal value and the approximate value yielded by the sampling process. We then extend the one-stage sampling process into multiple stages in a recursive manner, leading to a multi-stage (sampling-based) approximation algorithm for solving MDPs.

This paper is organized as follows. In Section 2, we give the necessary background and an intuitive description of the adaptive sampling algorithm, present a formal description of the algorithm, and discuss how to create an (approximate) receding horizon control (Hernández-Lerma and Lasserre, 1990) via the sampling algorithm to solve MDPs in an "on-line" manner in the context of "planning" for infinite horizon criteria. In Section 3, we provide the proofs for the convergence and the convergence rate of the worst-case bias, and in Section 4, we compare the algorithm with a nonadaptive sampling algorithm. In Section 5, we conclude this paper with some remarks.

2 Adaptive Sampling Algorithm

2.1 Background

Consider a finite horizon MDP $M = (X, A, P, R)$ with countable state space X, finite action space A with $|A| > 1$, nonnegative and bounded reward function R such that $R : X \times A \rightarrow \mathcal{R}^+$, and transition function P that maps a state and action pair to a probability distribution over X. We denote the probability of transitioning to state $y \in X$ when taking action a in state $x \in X$ by $P(x,a)(y)$. For simplicity, we assume that every action is admissible in every state.

Let Π be the set of all possible nonstationary Markovian policies $\pi = \{\pi_t | \pi_t : X \rightarrow A, t \geq 0\}$. Our goal is to estimate the optimal discounted total reward (thereby obtaining an (approximate) optimal policy) for horizon length H, discount factor γ, and initial state x_0. Defining the optimal reward-to-go value function for state x in stage i by

$$V_i^*(x) = \sup_{\pi \in \Pi} E\left[\sum_{t=i}^{H-1} \gamma^t R(x_t, \pi_t(x_t)) \middle| x_i = x\right], x \in X, 0 < \gamma \leq 1, i = 0, ..., H-1,$$

with $V_H^*(x) = 0$ for all $x \in X$ and x_t a random variable denoting the state at time t following policy π, we wish to estimate $V_0^*(x_0)$. Throughout the paper, we assume that γ is fixed. It is well-known (see, e.g., Bertsekas 1995) that V_i^* can be written recursively as follows: for all $x \in X$ and $i = 0, ..., H-1$,

$$\begin{aligned} V_i^*(x) &= \max_{a \in A}(Q_i^*(x,a)), \text{ where} \\ Q_i^*(x,a) &= R(x,a) + \gamma \sum_{y \in X} P(x,a)(y) V_{i+1}^*(y) \text{ with } V_H^*(x) = 0, x \in X. \end{aligned}$$

We remark that the work here can be extended to *Borel* state space with appropriate measure-theoretic arguments, and the assumption that we have the zero terminal reward function (for simplicity) can be relaxed with an arbitrary (bounded) terminal reward function.

Suppose we estimate $Q_i^*(x,a)$ by a sample mean $\hat{Q}_i(x,a)$ for each action $a \in A$, where

$$\hat{Q}_i(x,a) = R(x,a) + \gamma \frac{1}{N_{a,i}^x} \sum_{y \in S_a^x} \hat{V}_{i+1}^{N_{i+1}}(y), \tag{1}$$

where S_a^x is the *multiset* of (independently) sampled next states according to the distribution $P(x,a)$, and $|S_a^x| = N_{a,i}^x \geq 1$ for all $x \in X$ and such that $\sum_{a \in A} N_{a,i}^x = N_i$ for a fixed $N_i \geq |A|$ for all $x \in X$, and $\hat{V}_{i+1}^{N_{i+1}}(y)$ is an estimate of the unknown $V_{i+1}^*(y)$. Note that the number of next state samples depends on the state x, action a, and stage i. Suppose also that we estimate the optimal value of $V_i^*(x)$ by

$$\hat{V}_i^{N_i}(x) = \sum_{a \in A} \frac{N_{a,i}^x}{N_i} \hat{Q}_i(x,a).$$

This leads to the following recursion:

$$\hat{V}_i^{N_i}(x) := \sum_{a \in A} \frac{N_{a,i}^x}{N_i} \left(R(x,a) + \gamma \frac{1}{N_{a,i}^x} \sum_{y \in S_a^x} \hat{V}_{i+1}^{N_{i+1}}(y) \right), i = 0, ..., H-1,$$

with $\hat{V}_H^{N_H}(x) = 0$ for all $x \in X$ and any $N_H > 0$.

In the above definition, the total number of sampled (next) states is $O(N^H)$ with $N = \max_{i=0,...,H-1} N_i$, which is independent of the state space size. One approach is to select "optimal" values of $N_{a,i}^{x'}$ for $i = 0, ..., H-1$, $a \in A$, and $x' \in X$, such that the expected error between the values of $\hat{V}_0^{N_0}(x)$ and $V_0^*(x)$ is minimized, but this problem would be difficult to solve. So instead we seek the values of $N_{a,i}^{x'}$ for $i = 0, ..., H-1$, $a \in A$, and $x' \in X$ such that the expected difference is *bounded* as a function of $N_{a,i}^{x'}$ and N_i, $i = 0, ..., H-1$, and that the bound (from above and from below) goes to zero as N_i, $i = 0, ..., H-1$, go to infinity. We propose an "adaptive" allocation rule (sampling algorithm) that adaptively chooses which action to sample, updating the value of $N_{a,i}^{x'}$ as the sampling process proceeds, and achieves convergence such that as $N_i \rightarrow \infty$ for all $i = 0, .., H-1$, $E[\hat{V}_0^{N_0}(x)] \rightarrow V_0^*(x)$, and is efficient in the sense that the worst possible bias is bounded by a quantity that converges to zero at rate $O(\sum_i \frac{\ln N_i}{N_i})$ and the logarithmic bound in the numerator is achievable uniformly over time.

As mentioned before, the main idea behind the adaptive allocation rule is based on a simple interpretation of the regret analysis of the multi-armed bandit problem, a well-known model that captures the exploitation/exploration trade-off. An M-armed bandit problem is defined by random variables $K_{i,n}$ for $1 \leq i \leq M$ and $n \geq 1$, where successive plays of machine i yield "rewards" $K_{i,1}$,$K_{i,2}$,... which are independent and identically distributed according to an unknown but fixed distribution δ_i with unknown expectation μ_i. The rewards across machines are also independently generated. Let $T_i(n)$ be the number of times machine i has been played by an algorithm during the first n plays. Define the *expected regret* $\rho(n)$ of an algorithm after n plays by

$$\rho(n) = \mu^* n - \sum_{i=1}^{M} \mu_i E[T_i(n)] \text{ where } \mu^* := \max_i \mu_i.$$

Lai and Robbins (1985) characterized an "optimal" algorithm such that the best machine, which is associated with μ^*, is played exponentially more often than any other machine, at least asymptotically. That is, they showed that playing machines according to an (asymptotically) optimal algorithm leads to $\rho(n) = \Theta(\ln n)$ as $n \rightarrow \infty$ under mild assumptions on the reward distributions. Unfortunately, obtaining an optimal algorithm (proposed by Lai and Robbins) can sometimes be very difficult, so Agrawal (1995) derived a set of simple algorithms that achieve the asymptotic logarithmic regret behavior, using a form of *upper confidence bounds*. During the plays, we are temped to take the machine with the maximum current sample mean — exploitation. But the sample mean $\hat{\mu}_i(\bar{n})$ for the machine i is just an estimate that contains uncertainty, where $\bar{n}$ is the number of overall plays so far. To account for this, we add a function $\sigma_i(\bar{n})$ such that $\hat{\mu}_i(\bar{n}) - \sigma_i(\bar{n}) \leq \mu_i < \hat{\mu}_i(\bar{n}) + \sigma_i(\bar{n})$ with high probability, where $\hat{\mu}_i(\bar{n}) + \sigma_i(\bar{n})$ is the upper confidence bound (see Agrawal, 1995 for a substantial discussion). Then the width of the confidence bound gives us guidance for exploration. Indeed, the use of the upper confidence bound leads us to trade-off between exploitation and exploration, giving a criterion of which of the two between exploitation and exploration to be selected. Agrawal's algorithm is to choose the machine with the highest upper confidence bound at each play over time. For bounded rewards, Auer, Cesa-Bianchi, and Fischer (2002) propose simple upper confidence-bound based algorithms that achieve the logarithmic regret uniformly over time, rather than only asymptotically, and our sampling algorithm primarily builds on their results.

For an intuitive description of the allocation rule, consider first only the one-stage approximation. That is, we assume for now that we know $V_1^*(x)$ for all $x \in X$. To estimate $V_0^*(x)$, obviously we need to estimate $Q_0^*(x, a^*)$, where $a^* \in \arg\max_{a \in A}(Q_0^*(x, a))$. The search for a^* corresponds to the search for the best machine in the multi-armed bandit problem. We start by sampling each possible action once at x, which leads to the next state according to $P(x, a)$ and reward $R(x, a)$. We then iterate as follows (see **Loop** in Figure 1). The next action to sample is the one that achieves the maximum among the current estimates of $Q_0^*(x, a)$ plus its current upper confidence bound (see Equation (3)), where the estimate $\hat{Q}_0(x, a)$ is given by the immediate reward plus the *sample mean* of V_1^*-values at the *sampled next states that have been sampled so far* (see Equation (4)).

Among the N_0 samples for state x, $N_{a,0}^x$ denotes the number of samples at action a. If the sampling is done appropriately, we might expect that $\frac{N_{a,0}^x}{N_0}$ provides a good estimate of the probability that action a is optimal in state x. In the limit as $N_0 \to \infty$, we would expect $\frac{N_{a,0}^x}{N_0} \to 1$. Therefore, we use a weighted (by $\frac{N_{a,0}^x}{N_0}$) sum of the currently estimated value of $Q_0^*(x, a)$ over A to approximate $V_0^*(x)$ (see Equation (5)). Ensuring that the weighted sum concentrates on a^* as the sampling proceeds will ensure that in the limit the estimate of $V_0^*(x)$ converge to $V_0^*(x)$.

2.2 Algorithm description

We now provide a high-level description of the adaptive multi-stage sampling (AMS) algorithm to estimate $V_0^*(x)$ for a given state x in Figure 1. The inputs to AMS are a state $x \in X$, $N_i \geq |A|$, and stage i, and the output of AMS is $\hat{V}_i^{N_i}(x)$, the estimate of $V_i^*(x)$. Whenever we encounter $\hat{V}_k^{N_k}(y)$ for a state $y \in X$ and stage k in the **Initialization** and **Loop** portions of the AMS algorithm, we need to call AMS recursively (at Equation (2) and (5)). The initial call to AMS is done with $i = 0$, the initial state x, and N_0 and every sampling is done independently of the previously done samplings. To help understand how the recursive calls are made sequentially, in Figure 2, we graphically illustrate the sequence of calls with two actions and $H = 3$ for the **Initialization** portion.

The AMS algorithm is a recursive extension of the UCB1 algorithm given in Auer, Cesa-Bianchi, and Fischer (2002) in the context of the MDP framework. It is based on the index-based policy of Agrawal (1995), where the index for an action is given by the sum of the current estimate of the true Q-value for the action plus a term that relates the size of the upper confidence bound.

Adaptive Multi-stage Sampling (AMS)

- **Input:** a state $x \in X$, $N_i \geq |A|$, and stage i. **Output:** $\hat{V}_i^{N_i}(x)$.
- **Initialization:** Sample each action $a \in A$ sequentially once at state x and set

$$\hat{Q}_i(x,a) = \begin{cases} 0 \text{ if } i = H \text{ and go to } \textbf{Exit} \\ R(x,a) + \gamma \hat{V}_{i+1}^{N_{i+1}}(y) \text{ if } i \neq H, \end{cases} \tag{2}$$

  where y is the sampled next state with respect to $P(x,a)$, and set $\bar{n} = |A|$.

- **Loop:** Sample sequentially each action a^* that achieves

$$\max_{a \in A} \left(\hat{Q}_i(x,a) + \sqrt{\frac{2 \ln \bar{n}}{N_{a,i}^x}} \right), \tag{3}$$

  where $N_{a,i}^x$ is the number of times action a has been sampled so far, and $\bar{n}$ is the overall number of samples done so far for this stage, and $\hat{Q}_i$ is defined by

$$\hat{Q}_i(x,a) = R(x,a) + \gamma \frac{1}{N_{a,i}^x} \sum_{y \in S_a^x} \hat{V}_{i+1}^{N_{i+1}}(y), \tag{4}$$

  where S_a^x is the set of sampled next states so far with $|S_a^x| = N_{a,i}^x$ with respect to the distribution $P(x,a)$.

  – Update $N_{a^*,i}^x \leftarrow N_{a^*,i}^x + 1$ and $S_{a^*}^x \leftarrow S_{a^*}^x \cup \{y'\}$, where y' is the newly sampled next state by a^*.

  – Update $\hat{Q}_i(x,a^*)$ with the $\hat{V}_{i+1}^{N_{i+1}}(y')$ value.

  – $\bar{n} \leftarrow \bar{n} + 1$. If $\bar{n} = N_i$, then exit **Loop**.

- **Exit:** Set $\hat{V}_i^{N_i}(x)$ such that

$$\hat{V}_i^{N_i}(x) = \begin{cases} \sum_{a \in A} \frac{N_{a,i}^x}{N_i} \hat{Q}_i(x,a) \text{ if } i = 0, ..., H-1 \\ 0 \text{ if } i = H. \end{cases} \tag{5}$$

  and return $\hat{V}_i^{N_i}(x)$.

Figure 1: Adaptive multi-stage sampling algorithm (AMS) description

The running time-complexity of the AMS algorithm is $O((|A|N)^H)$ with $N = \max_i N_i$. To see this, let M_i be the number of recursive calls to make to compute $\hat{V}_i^{N_i}$ at the *worst* case. At stage i, AMS makes $|A|M_{i+1}$ recursive calls in **Initialization** and $|A|N_iM_{i+1}$ calls in **Loop** at the worst

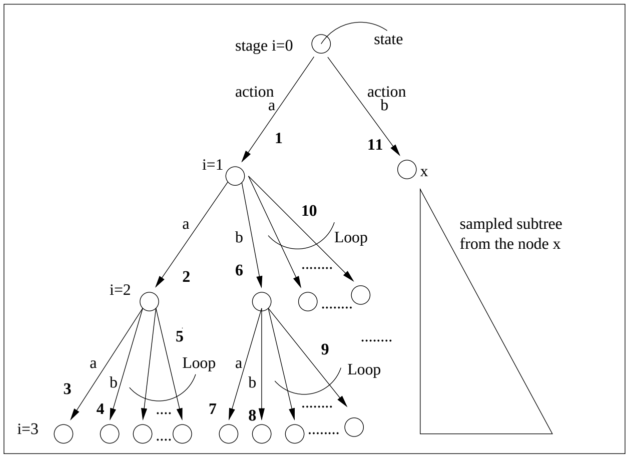

Figure 2: Graphical illustration of the sequence of the recursive calls made in **Initialization** of the AMS algorithm. Each circle corresponds to a state and each arrow with noted action signifies a sampling (and a recursive call). The bold-face number near each arrow is the sequence number for the recursive calls made. For simplicity, the entire **Loop** process is signified by one call number.

case. Therefore, $M_i = (|A| + |A|N_i)M_{i+1}$ so that $M_0 = O((|A| + |A|N)^H) = O((|A|N)^H)$. In contrast, backward induction has $O(H|A||X|^2)$ running time-complexity (see, e.g., Blondel 2000). Therefore, the main benefit of AMS is independence from the state space size, but this comes at the expense of exponential (versus linear, for backwards induction) dependence on both the action space and the horizon length.

2.3 Creating an on-line stochastic policy

Once armed with an algorithm that estimates the optimal value for finite horizon problems, we can create a nonstationary stochastic policy in an on-line manner in the context of "planning" (see, e.g., Kearns, Mansour, and Ng 2001). Suppose at time $t \geq 0$, we are at state $x \in X$. We evaluate each action's utility as follows:

$$R(x,a) + \gamma \frac{1}{N_t} \sum_{y \in S_a^x} \hat{V}_{t+1}^{N_{t+1}}(y), a \in A, \tag{6}$$

where we apply the AMS algorithm at the sampled next states for the stage $t+1$. We simply take the action that achieves the maximum utility. We remark that the use of common random numbers (see, e.g., Law and Kelton 2000) across actions in the utility measures given by Equation (6) should reduce the variance in the spirit of "differential training" in the rollout algorithm (Bertsekas 1997).

If we replace the horizon $H-1$ by $t+H$ in the definition of $\hat{V}_{t+1}^{N_{t+1}}(y)$ in the above equation (6), the resulting stochastic policy yields an (approximate) receding H-horizon control (Hernández-Lerma and Lasserre 1990) for the infinite horizon problem.

3 Convergence Analysis

In this section, we prove the convergence of the AMS algorithm and show that the worst possible bias converges to zero at rate $O\left(\sum_{i=0}^{H-1} \frac{\ln N_i}{N_i}\right)$.

Theorem 3.1 *Let $R_{\max} = \sup_{x,a} R(x,a)$ and assume that $R_{\max} \leq \frac{1}{H}$. Suppose AMS is run with the input N_i for stage $i = 0, ..., H-1$ and an arbitrary initial state $x \in X$. Then*

$$\lim_{N_0\to\infty} \lim_{N_1\to\infty} \cdots \lim_{N_{H-1}\to\infty} E[\hat{V}_0^{N_0}(x)] = V_0^*(x).$$

Proof of Theorem 3.1:

We start with a convergence result for the one-stage approximation. Consider the following one-stage sampling algorithm (OSA) in Figure 3 with a *stochastic value function* U defined over X. $U(x)$ for $x \in X$ is a *nonnegative random variable* with *unknown* distribution and bounded above for all $x \in X$. We will denote $U(x)$ as a (random) sample from the unknown distribution associated with $U(x)$. As before, every sampling is done independently and we are assuming that there is a black box that returns $U(x)$ once x is given to the black box. Let

$$U_{\max} = \sup_{x,a} \left(R(x,a) + \gamma \sum_{y\in X} P(x,a)(y) E[U(y)] \right),$$

and assume for the moment that $U_{\max} \leq 1$.

We state a key lemma that will be used to prove the convergence of the AMS algorithm.

Lemma 3.1 *Given a stochastic value function U defined over X with $U_{\max} \leq 1$, suppose we run OSA with the input n. Define for all $x \in X$,*

$$V(x) = \max_{a\in A} \left(R(x,a) + \gamma \sum_{y\in X} P(x,a)(y) E[U(y)] \right).$$

Then, for all $x \in X$,

$$E[\tilde{V}^n(x)] \to V(x) \text{ as } n \to \infty.$$

Proof of Lemma 3.1:

Fix a state $x \in X$ and index each action in $|A|$ by numbers from 1 to $|A|$. Consider an $|A|$-armed bandit problem where each a is a gambling machine. Successive plays of machine a yield "bandit rewards" which are independent and identically distributed according to an unknown distribution δ_a with unknown expectation

$$Q(x,a) = R(x,a) + \gamma \sum_{y\in X} P(x,a)(y) E[U(y)],$$

and are independent across machines or actions.

One-stage Sampling Algorithm (OSA)

- **Input:** a state $x \in X$ and $n \geq |A|$.
- **Initialization:** Sample each action $a \in A$ once at state x and set
$$\tilde{Q}(x,a) = R(x,a) + \gamma U(y),$$
where y is the sampled next state with respect to $P(x,a)$, and set $\bar{n} = |A|$.
- **Loop:** Sample each action a^* that achieves
$$\max_{a \in A} \left(\tilde{Q}(x,a) + \sqrt{\frac{2 \ln \bar{n}}{T_a^x(\bar{n})}} \right),$$
where $T_a^x(\bar{n})$ is the number of times action a has been sampled so far at state x, $\bar{n}$ is the overall number of samples done so far, and $\tilde{Q}$ is defined by
$$\tilde{Q}(x,a) = R(x,a) + \gamma \frac{1}{T_a^x(\bar{n})} \sum_{y \in \Lambda_a^x} U(y),$$
where Λ_a^x is the set of sampled next states so far with $|\Lambda_a^x| = T_a^x(\bar{n})$ with respect to the distribution $P(x,a)$.
  - Update $T_{a^*}^x(\bar{n}) \leftarrow T_{a^*}^x(\bar{n}) + 1$ and $\Lambda_{a^*}^x \leftarrow \Lambda_{a^*}^x \cup \{y'\}$, where y' is the newly sampled next state by a^*.
  - Update $\tilde{Q}(x,a^*)$ with $U(y')$.
  - $\bar{n} \leftarrow \bar{n} + 1$. If $\bar{n} = n$, then exit **Loop**.
- **Exit:** Set $\tilde{V}^n$ such that
$$\tilde{V}^n(x) = \sum_{a \in A} \frac{T_a^x(n)}{n} \tilde{Q}(x,a). \tag{7}$$

Figure 3: One-stage sampling algorithm (OSA) description

The term $T_a^x(n)$ signifies the number of times machine a has been played (or action a has been sampled) by OSA during the n plays. Define the *expected regret* $\rho(n)$ of OSA after n plays by

$$\rho(n) = V(x)n - \sum_{a=1}^{|A|} Q(x,a) E[T_a^x(n)], \text{ where } V(x) = \max_{a \in A} Q(x,a).$$

Applying Theorem 1 from Auer, Cesa-Bianchi, and Fischer (2002) gives the following bound on $\rho(n)$:

Theorem 3.2 *For all $|A| > 1$, if OSA is run on $|A|$-machines having arbitrary bandit reward distribution $\delta_1, ..., \delta_{|A|}$ with $U_{\max} \leq 1$,*

$$\rho(n) \leq \sum_{a:Q(x,a)<V(x)} \left[\frac{8 \ln n}{V(x) - Q(x,a)} + (1 + \frac{\pi^2}{3})(V(x) - Q(x,a)) \right],$$

where $Q(x,a)$ is the expected value of bandit rewards with respect to δ_a.

See Auer, Cesa-Bianchi, and Fischer (2002) for a proof of the above theorem. Observe that $\max_a(V(x) - Q(x,a)) \leq U_{\max}$. Let $\phi(x) = \{a|Q(x,a) < V(x), a \in A\}$, i.e., the set of nonoptimal actions for x. Define $\alpha(x)$ for $\phi(x) \neq \emptyset$ such that

$$\alpha(x) = \min_{a\in\phi(x)} (V(x) - Q(x,a)) \tag{8}$$

and note that $0 < \alpha(x) \leq U_{\max}$. Define

$$\tilde{V}(x) = \sum_{a=1}^{|A|} \frac{T_a^x(n)}{n} Q(x,a).$$

Applying Theorem 3.2, we have

$$0 \leq V(x) - E[\tilde{V}(x)] = \frac{\rho(n)}{n} \leq \frac{8(|A|-1)\ln n}{n\alpha(x)} + (1 + \frac{\pi^2}{3}) \cdot \frac{(|A|-1)U_{\max}}{n}. \tag{9}$$

Note also that $\rho(n) = 0$ if $\phi(x) = \emptyset$.

From the definition of $\tilde{V}^n(x)$ given by Equation (7), it follows that

$$\begin{aligned} V(x) - E[\tilde{V}^n(x)] &= V(x) - E[\tilde{V}(x) - \tilde{V}(x) + \tilde{V}^n(x)] \\ &= V(x) - E[\tilde{V}(x)] + E\left[\sum_{a\in A} \frac{T_a^x(n)}{n} \Big(Q(x,a) - \tilde{Q}(x,a)\Big)\right]. \end{aligned} \tag{10}$$

Letting $n \to \infty$, the first term $V(x) - E[\tilde{V}(x)]$ is bounded by zero from above with convergence rate of $O(\frac{\ln n}{n})$ by Equation (9). We show now that the second expectation term is zero.

First observe that $T_a^x(n)$ for every finite n is a *stopping time* (see, e.g., Ross 1995, p.104) with $E[T_a^x(n)] \leq n < \infty$. Let $\mu_a = \sum_{y\in X} P(x,a)(y)E[U(y)]$.

$$\begin{aligned} E\left[\sum_{a\in A} \frac{T_a^x(n)}{n} \Big(Q(x,a) - \tilde{Q}(x,a)\Big)\right] &= E\left[\sum_{a\in A} \frac{T_a^x(n)}{n} \left(R(x,a) + \gamma\mu_a - R(x,a) - \gamma\frac{1}{T_a^x(n)} \sum_{y\in\Lambda_a^x} U(y)]\right)\right] \\ &= \frac{\gamma}{n}\left(\sum_{a\in A} E[T_a^x(n)]\mu_a\right) - \frac{\gamma}{n}E\left[\sum_{a\in A}\left(\sum_{y\in\Lambda_a^x} U(y)\right)\right] \\ &= \frac{\gamma}{n}\left(\sum_{a\in A} E[T_a^x(n)]\mu_a\right) - \frac{\gamma}{n}\sum_{a\in A} E\left[\sum_{y\in\Lambda_a^x} U(y)\right] \\ &= 0 \text{ by applying Wald's equation.} \end{aligned}$$

Since

$$V(x) - E[\tilde{V}^n(x)] = V(x) - E[\tilde{V}(x)],$$

the convergence follows directly from Equation (9).

Therefore, because x was chosen arbitrarily, we have that for all $x \in X$,

$$E[\tilde{V}^n(x)] \to V(x) \text{ as } n \to \infty,$$

which concludes the proof of Lemma 3.1. ∎

We now return to the AMS algorithm. From the definition of $\hat{V}_{H-1}^{N_{H-1}}$,

$$\begin{aligned}\hat{V}_{H-1}^{N_{H-1}}(x) &= \sum_{a\in A} \frac{N_{a,H-1}^x}{N_{H-1}} \left(R(x,a) + \gamma \frac{1}{N_{a,H-1}^x} \sum_{y \in S_a^x} \hat{V}_H^{N_H}(y) \right) \\ &\le \sum_{a\in A} \frac{N_{a,H-1}^x}{N_{H-1}} \left(R_{\max} + \gamma \cdot 0\right) = R_{\max}, x \in X.\end{aligned}$$

Similarly for $\hat{V}_{H-2}^{N_{H-2}}$, we have that

$$\begin{aligned}\hat{V}_{H-2}^{N_{H-2}}(x) &= \sum_{a\in A} \frac{N_{a,H-2}^x}{N_{H-2}} \left(R(x,a) + \gamma \frac{1}{N_{a,H-2}^x} \sum_{y \in S_a^x} \hat{V}_{H-1}^{N_{H-1}}(y) \right) \\ &\le \sum_{a\in A} \frac{N_{a,H-2}^x}{N_{H-2}} \left(R_{\max} + \gamma R_{\max}\right) = R_{\max}(1+\gamma), x \in X.\end{aligned}$$

Continuing this backwards, we have for all $x \in X$ and $i = 0, ..., H-1$,

$$\hat{V}_i^{N_i}(x) \le R_{\max} \sum_{j=0}^{H-i-1} \gamma^j \le R_{\max}(H-i) \le 1,$$

where the last inequality comes from the assumption that $R_{\max} H \le 1$.

Therefore, from Lemma 3.1 with $U_{\max} = R_{\max}(H-i) \le 1$, we have for $i = 0, ..., H-1$, and for arbitrary $x \in X$,

$$E[\hat{V}_i^{N_i}(x)] \overset{N_i \to \infty}{\longrightarrow} \max_{a \in A} \left(R(x,a) + \gamma \sum_{y \in X} P(x,a)(y) E[\hat{V}_{i+1}^{N_{i+1}}(y)] \right).$$

But for arbitrary $x \in X$, because $\hat{V}_H^{N_H}(x) = V_H^*(x) = 0, x \in X$,

$$E[\hat{V}_{H-1}^{N_{H-1}}(x)] \overset{N_{H-1} \to \infty}{\longrightarrow} V_{H-1}^*(x),$$

which in turn leads to $E[\hat{V}_{H-2}^{N_{H-2}}(x)] \to V_{H-2}^*(x)$ as $N_{H-2} \to \infty$ for arbitrary $x \in X$, and by an inductive argument, we have that

$$\lim_{N_0 \to \infty} \lim_{N_1 \to \infty} \cdots \lim_{N_{H-1} \to \infty} E[\hat{V}_0^{N_0}(x)] = V_0^*(x) \text{ for all } x \in X,$$

which concludes the proof of Theorem 3.1. ∎

We now argue that the worst possible bias by AMS is bounded by a quantity that converges to zero at rate $O\left(\sum_{i=0}^{H-1} \frac{\ln N_i}{N_i}\right)$. Let $B(X)$ be the space of real-valued bounded measurable functions on X endowed with the supremum norm $\|\Phi\| = \sup_x |\Phi(x)|$ for $\Phi \in B(X)$. We define an operator $T : B(X) \to B(X)$ as

$$T(\Phi)(x) = \max_{a\in A}\left\{R(x,a) + \gamma \sum_{y\in X} P(x,a)(y)\Phi(y)\right\}, \Phi \in B(X), x \in X. \tag{11}$$

Define $\Psi_i \in B(X)$ such that $\Psi_i(x) = E[\hat{V}_i^{N_i}(x)]$ for all $x \in X$ and $i = 0, ..., H-1$ and $\Psi_H(x) = V_H^*(x) = 0, x \in X$. In the proof of Lemma 3.1 (see Equation (10)), we showed that for $i = 0, ..., H-1$,

$$T(\Psi_{i+1})(x) - \Psi_i(x) \leq O\left(\frac{\ln N_i}{N_i}\right), x \in X.$$

Therefore, we have

$$T(\Psi_1)(x) - \Psi_0(x) \leq O\left(\frac{\ln N_0}{N_0}\right), x \in X. \tag{12}$$

and

$$\Psi_1(x) \geq T(\Psi_2)(x) - O\left(\frac{\ln N_1}{N_1}\right), x \in X. \tag{13}$$

Applying the T-operator to both sides of Equation (13) and using the monotonicity property of T (see, e.g., Bertsekas 1995), we have

$$T(\Psi_1)(x) \geq T^2(\Psi_2)(x) - O\left(\frac{\ln N_1}{N_1}\right), x \in X. \tag{14}$$

Therefore, combining Equation (12) and (14) yields

$$T^2(\Psi_2)(x) - \Psi_0(x) \leq O\left(\frac{\ln N_0}{N_0} + \frac{\ln N_1}{N_1}\right), x \in X.$$

Repeating this argument yields

$$T^H(\Psi_H)(x) - \Psi_0(x) \leq O\left(\sum_{i=0}^{H-1} \frac{\ln N_i}{N_i}\right), x \in X. \tag{15}$$

Observe that $T^H(\Psi_H)(x) = V_0^*(x), x \in X$. Rewriting Equation (15), we finally have

$$V_0^*(x) - E[\hat{V}_0^{N_0}(x)] \leq O\left(\sum_{i=0}^{H-1} \frac{\ln N_i}{N_i}\right), x \in X,$$

and we know that $V_0^*(x) - E[\hat{V}_0^{N_0}(x)] \geq 0, x \in X$. Therefore, it implies that the worst possible bias is bounded by the quantity that converges to zero at rate $O\left(\sum_{i=0}^{H-1} \frac{\ln N_i}{N_i}\right)$.

We remark that we can relax the assumption $R_{\max} \leq \frac{1}{H}$, by a normalization of the given reward function. The upper bound in Theorem 3.2 for $\rho(n)$ needs to be modified with a different bounded constant from $1+\frac{\pi^2}{3}$, which can be achieved by the Hoeffding inequality with support in $[0, R_{\max}H]$ rather than in $[0,1]$. Therefore, the assumption of the support in $[0,1]$ is not crucial (Cesa-Bianchi and Fisher 1998).

4 Comparison with a Nonadaptive Algorithm

Consider the following recursive definition: given fixed $N > 0$,

$$\bar{V}_i^N(x) = \max_{a \in A} \left(R(x,a) + \frac{1}{N} \sum_{y \in \Omega_a^x} \bar{V}_{i+1}^N(y) \right), i = 0, ..., H-1, x \in X,$$

with $\bar{V}_H^N(x) = 0$ for all $x \in X$, where Ω_a^x is the multiset of sampled states with respect to $P(x,a)$ and $|\Omega_a^x| = N$ for all $a \in A$ and $x \in X$.

The above recursive definition immediately suggests the following *nonadaptive* multi-stage sampling (NMS) algorithm. NMS creates a random sample-path tree having the depth of H and branching factor of $N|A|$ in forward manner, where N is the fixed number of next states to be sampled from each sampled state in the sample-path tree for each action, and then in backward manner, the estimate value of V_0^*-value or $\bar{V}_0^N$-value is computed recursively (see Kearns, Mansour, and Ng 2001 for detailed description and a performance analysis of NMS for infinite horizon discounted criterion). Note that the running time-complexity of NMS is $O\left((|A|N)^H\right)$, which is similar to that of AMS at the *worst case*, and NMS is asymptotically unbiased in the sense that as $N \rightarrow \infty$, $E[\bar{V}_0^N(x)] \rightarrow V_0^*(x)$ simply by the law of large numbers. This motivates us to compare the convergence rates between AMS and NMS.

Via the Hoeffding inequality, it is straightforward to establish that for all $x \in X$ and $\epsilon > 0$,

$$\Pr\left\{|V_0^*(x) - \bar{V}_0^N(x)| > \epsilon\right\} \leq 2(N|A|)^H e^{-2N\epsilon^2/H^2}$$

with assumption that $R_{\max}H \leq 1$. (See Lemma 3 and 4 in (Kearns, Mansour, and Ng 2001) with appropriate modifications in the context of the discounted finite-horizon total reward criterion.) Because application of the Hoeffding inequality to obtain the expected performance error does not provide any useful information, we use the upper bound Markov inequality (see, e.g., Hofri 1995, p.574): for a nonnegative bounded random variable K, for any $\epsilon > 0$,

$$EK \leq \sup K \cdot \Pr\{K > \epsilon\} + \epsilon.$$

It follows that with $\epsilon > 0$,

$$E|V_0^*(x) - \bar{V}_0^N(x)| \leq R_{\max}H \cdot 2(N|A|)^H e^{-2N\epsilon^2/H^2} + \epsilon.$$

Therefore, to make the expected error go to zero, we need to select an arbitrarily close to zero value of ϵ with $N \rightarrow \infty$. However, the choice of ϵ will make the exponential term in the denominator almost constant even with a very large N. Therefore, we expect that the convergence rate of the nonadaptive algorithm will be much slower than the convergence rate of AMS even with a value of ϵ not that close to zero in practice (e.g., due to the exponential dependence on the horizon size in the numerator) even though the main benefit of the NMS algorithm would be independence from the state space like AMS.

5 Concluding Remarks

To the best of our knowledge, this is the first work applying the theory of the multi-armed bandit problem to derive a provably convergent algorithm for solving general finite-horizon MDPs. The closest related work is probably that of Agrawal, Teneketzis, and Anatharam (1989), who considered a controlled Markov chain problem with finite state and action spaces, where transition probabilities and initial distribution are parameterized by an unknown parameter belonging to some known finite parameter space and each Markov chain induced from each fixed parameter is irreducible and aperiodic. Defining a loss function based on the regret of Lai and Robbins (1985), they provide an "asymptotically efficient" adaptive but complex control scheme that works well for all parameters such that the loss associated with the control scheme is equal to the lower bound on the loss function asymptotically (as we apply the scheme over infinite number of time steps). The adaptiveness comes from the use of the multi-armed bandit theory for the stationary control laws. In other words, the arm corresponds to a particular stationary law or policy, but not a particular action in the action space. We believe that extending the AMS algorithm within the context of Agrawal, Teneketzis, and Anatharam (1989) is not difficult, achieving a uniform rather than asymptotic result over time-steps.

We assumed without loss of generality that $|A| > 1$, because problems for which $|A| = 1$, or the number of the admissible actions for some states is one, can be solved by the following transformation of $M = (X, A, P, R)$ to an equivalent $M' = (X', A', R', P')$ as follows. We augment X with an extra state $\bar{x}$ and A with an extra action $\bar{a}$ such that $X' = X \cup \{\bar{x}\}$ and $A' = A \cup \{\bar{a}\}$. The state transition function P' is defined such that $P'(x,a)(y) = P(x,a)(y)$ for all $x, y \in X$ and $a \in A$, and $P'(\bar{x},a)(\bar{x}) = 1$ for all $a \in A'$, and for all $x \in X$, $P'(x,a)(\bar{x}) = 1$ if $a = \bar{a}$ and 0 if not. The reward function R' is defined such that $R'(x,a) = R(x,a)$ for all $x \in X$ and $a \in A$, and $R'(\bar{x},a) = 0$ for all $a \in A'$. Note that $\bar{x}$ is just a sink state that is only reachable by the action $\bar{a}$ from all states in X and $\bar{a}$ is always a suboptimal action at each state. It is left for the reader to check that the transformed MDP M' gives the same optimal action at each state in X as M and the same optimal value at each state in X as M.

For the actual implementation of the AMS algorithm, we can use the same value $N = N_i$, $i = 0, ..., H-1$, and we can improve the running time-complexity of AMS in heuristic ways as follows. Given an MDP $M = (X, A, P, R)$, consider the transformation of M into $M' = (X', A', P', R')$ as discussed in the previous paragraph. Suppose that we add the following structure to M' on the state transition function P'. $P'(x,a)(\bar{x})$ is arbitrarily close to 0 (instead of 0) for all $x \in X$ and $a \in A$ with a proper normalization of $P(x,a)(y)$ for $y \in X$. That is, each state $x \in X$ can reach the sink state with close to zero probability. Then solving the newly defined MDP is *almost* equivalent to the original MDP M. This speculation suggests that when we apply AMS for a state $x \in X$, in the **Initialization** step, we just set $\hat{Q}_i(x,a) = R(x,a)$ for $i = 0, ..., H-1$, pretending that the sampled next state is the sink state, eliminating $O(N^H)$ computations. Furthermore, if $\gamma \neq 1$, we can set, e.g., $N_0 = n$ and $N_i = \lfloor \gamma^i n \rfloor$ for $i \geq 1$ heuristically, incorporating the discounting nature.

We can extend the AMS algorithm to include the case where the reward function is random. The AMS algorithm would essentially remain identical, except that sampling would now include both the next state and the one-stage reward. However, the convergence proof is likely to require more technical manipulations. Furthermore, the assumption of bounded rewards can be relaxed by using the result in Agrawal (1995). Even though the AMS algorithm will converge too in this case, unfortunately, we lose the property of the uniform logarithmic bound so that the convergence rate is expected to be very slow.

We can use the AMS algorithm to approximate the optimal infinite horizon discounted average reward and the infinite horizon average reward under an ergodicity assumption by (approximately) solving a finite-horizon MDP. Deriving an expected error bound for each case is straightforward.

Earlier work of Cesa-Bianchi and Fischer (1998) proposed several algorithms that achieve the regret bounds of the form $c_1 + c_2 \log n + c_3 \log^2 n$, where n is the total number of plays and c_i's are positive constants not depending on n. These algorithms might also be used to create adaptive sampling algorithms for solving MDPs. However, those algorithms have the drawback that we need to know the exact value of $\alpha(x)$ for a given state x under the assumption that not all of the actions are optimal, which is difficult to obtain in advance. This holds also for other algorithms studied in Auer, Cesa-Bianchi, and Fischer (2002).

The alert reader might wonder what happens if we replace the weighted sum of the Q-value estimates in Equation (5) by the maximum of the estimates instead of the weighted sum. We expect that the resulting algorithm will also converge to the true optimal value. However, to analyze this we need to know how the distribution of the maximum of the estimates changes while running the algorithm, which would be very difficult. The proof of the convergence the resulting algorithm is an open problem.

Note 1 Throughout the paper, the notation O used in the sense that for given two functions f and g, $f(n) = O(g(n))$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c$ for some constant $c > 0$, and the notation Θ is used in that there exist positive constants c_1, c_2, and n_0 such that $0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ for all $n \geq n_0$ (Cormen, Leiserson, and Rivest 1990). The O and Θ-notations are often called asymptotic upper bound and asymptotically tight bound, respectively for the asymptotic running time of an algorithm.

Acknowledgements This work was supported in part by the National Science Foundation under Grant DMI-9988867, in part by the Air Force Office of Scientific Research under Grant F496200110161, and in part by the Department of Defense under Contract MDA 90499C2521.

References

Agrawal, R. 1995. Sample mean based index policies with $O(\log n)$ regret for the multi-armed bandit problem. Advances in Applied Probability. 27, 1054–1078.

Agrawal, R., Teneketzis, D., and Anantharam, V. 1989. Asymptotically efficient adaptive allocation schemes for controlled Markov chains: finite parameter space. IEEE Trans. on Automatic Control. 34, 1249–1259.

Auer, P., Cesa-Bianchi, N., and Fisher, P. 2002. Finite-time analysis of the multiarmed bandit problem. Machine Learning. 47, 235–256.

Bertsekas, D. P. 1995. Dynamic Programming and Optimal Control, Volumes 1 and 2. Athena Scientific.

Bertsekas, D. P. 1997. Differential training of rollout policies. Proc. 35th Allerton Conference on Communication, Control, and Computing.

Blondel, V. D., and Tsitsiklis, J. 2000. A survey of computational complexity results in systems and control. Automatica. 36, 1249–1274.

Cesa-Bianchi, N., and Fisher, P. 1998. Finite-time regret bounds for the multiarmed bandit problem. Proc. 15th Int. Conf. on Machine Learning.

Cormen, T. H., Leiserson, C. E., and Rivest, R. L. 1990. Introduction to Algorithms. MIT Press.

Hernández-Lerma, O., and Lasserre, J. B. 1990. Error bounds for rolling horizon policies in discrete-time Markov control processes. IEEE Trans. on Automatic Control. 35, 1118–1124.

Hofri, M. 1995. Analysis of Algorithms. Oxford University Press.

Kearns, M., Mansour, Y., and Ng, A. Y. 2001. A sparse sampling algorithm for near-optimal planning in large Markov decision processes. Machine Learning. 49, 193–208.

Lai, T., and Robbins, H. 1985. Asymptotically efficient adaptive allocation rules. Advances in Applied Mathematics. 6, 4–22.

Law, A. M., and Kelton, W.D. 2000. Simulation Modeling and Analysis. 3rd ed. McGraw-Hill, New York.

Ross, S. 1995. Stochastic Process. 2nd ed. John Wiley & Sons.